# DEVELOPMENT OF A P. C. CONTROLLED MOBILE ROBOT

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of

**MASTER OF TECHNOLOGY**

by
VENKATARAMAN SUBRAMANIYAM

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

AUGUST, 1988

ME-1988-M-SUB-DEV

CERTIFICATE

This is to certify that the thesis entitled "Development of a P.C controlled Mobile Robot " by Mr. Venkataraman Subramaniyam has been carried out under our supervision and has not been submitted elsewhere for a degree.

[illegible]

(Dr. Amitabha Ghosh)

Professor,

[illegible]

(Dr. Ashok Kumar Mallik)

Professor,

Department of Mechanical Engineering,

Indian Institute Of Technology, Kanpur.

## ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude to my guides fc allowing me to work with full freedom. I also wish to thank or and all of the Manufacturing Science Lab. and Vibration and Contrc Lab., particularly Mr. R.M. Jha, Mr. O.P. Bajaj, Mr. Bharthia, Mı Sharma, Mr. Singh and Mr. Majumdar. No doubt the other suppoı personnel also helped me a lot. Thanks are due to all my frien who let me execute my dual responsility as President Hall IV and a student in particular Anbarasu Special thanks are due to B Kapoor, Vasudev, Huliyal, Bannur, Sanjanwala and Katiyar. T SVRITES no doubt provided the required boosters for a nostalg existance with added dose from the TAMIL Group.. Most electric students were involved in some way or the other to develop t system. I wish to record here my thanks to Ramakrish <YS>(R.E.,ACES), Mr. Raghuram,Subbu and Dr. A.Joshi.

## ABSTRACT

This project aims at increasing the versatality of the earlier model of the vehicle. The changes in the mechanical design of the system results in improvement of the effeciency and stability while removing the difficulty of entanglement of cable used for communication. The vehicle is fitted with a Manipulator (SIR-1) and is interfaced with a personal computer through RS-232 serial interface. The program executed by the host is used to avoid predefined obstacles in the work space. Visibility Graph method is used to map all the possible via points. The data regarding the number of steps involved is then transferred to the system (vehicle) controller's memory for further action.

CONTENTS

Chapter 1 Introduction

1.1 Introduction 1
1.2 Previous work done 2
1.3 Objectives of the present work 6
1.4 Scope of the present work 7

Chapter 2 Hardware Design

2.1 Introduction 9

2.2 Design of Various Mechanical Components 9
- 2.2.1 Transmission system 9
- 2.2.2 Castor Wheel Fixtures 11
- 2.2.3 Motor Housing 16
- 2.2.4 Other Modifications 16

2.3 Wire Winding/Unwinding Mechanism
- 2.3.1 Drum 19
- 2.3.2 Support Rollers 21
- 2.3.3 Sensor Roller 21

2.4 Electrical Hardware 25

Chapter 3 Software Design

3.1 Introduction 31

3.2 Algorithms
- 3.2.1 Visibility Method 33
- 3.2.2 Short Path Search 34
- 3.2.3 Microprocessor Algorithm 34

3.3 Software P.C. Based
- 3.3.1 Visibility Method 35
- 3.3.2 Dijkstra's Algorithm 36
- 3.3.3 Location Retrival 37

3.4 System Software (8Ø85 Based)
- 3.4.1 Controller Program 37
- 3.4.2 Service Routines for Motors 38
- 3.4.3 Location Retrival 38

Chapter 4 Results of Experiments 4Ø

Chapter 5 Scope for further work

| | | |
|---|---|---|
| | 5.1 Introduction | 41 |
| | 5.2.1 Communication System between station and Robot | 43 |
| | 5.2.2 Vision System | 44 |
| APPENDIX 1 | Mathematical Analysis | 45 |
| APPENDIX 2 | Degree of Complexity | 47 |
| APPENDIX 3 | Assembly language (8Ø85) Program | 48 |
| REFERENCES | | 55 |

# CHAPTER 1

# INTRODUCTION

## 1.1 INTRODUCTION

From the beginning of civilization, man has always tried to build devices which reduce his physical and repeatative actions. Thus ROBOT, or Man's slave as it translates, was developed. The limitation of a slave (Robot) in a fixed location makes one think of giving it greater range of action by providing mobility to it. The systems known as "TELECHIRS" or TELEOPERATOR", first proposed by J. W. Clark in 1962 tends to satisfy this desire of man. The basic need for a telechir is to do remotely what man would have done had he been in such an environment. The human operator controls the motion of the end effector through a linkage chain. The potential health hazzards caused to man by various work environments in particular one with nuclear radiation necessiates the use of telechirs [1]. In addition to application in nuclear engineering the mobile robot is widely used in the fields of space technology, mining, and underwater operations [2].

A wheeled vehicle, such as a motor vehicle with firm tyres, represents an ideal system with very little consumption of energy for its locomotion. However such a vehicle can work only on surfaces which are reasonably smooth. The system has very little energy wastage since there is practically no reciprocating or sliding motion. This discrete wheel system is not meant to climb gradients which are too large.

A teleoperator is not meant to be intelligent and man's intervention is expected any time. In practice teleoperator is used in either of the two modes, namely

(i)with the help of a human operator's vision to guide and help in its navigation or-

(ii)with the help of a sophisicated vision system with cameras and sensors to do the sensing and to provide the required insight to perform a given task. Ideally, a self contained robot completely independent of any human intervention is sought. However, such a robot involves a lot of preconsiderations for its operation.

## 1.2 PREVIOUS WORK DONE

Different mobile systems have been developed by various research centers the world over. Some of the more significant contributions are listed below :

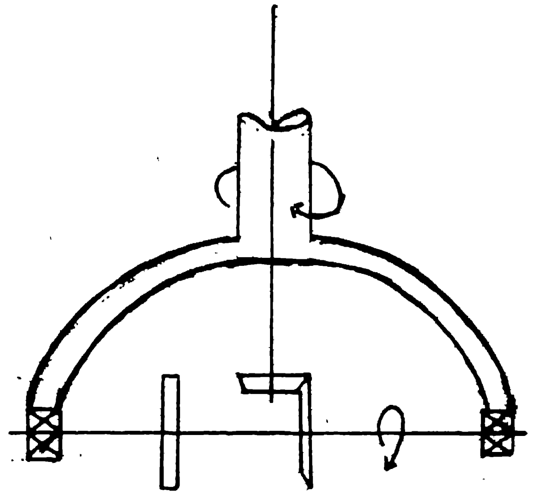
(i) SHAKEY - This was perhaps the first effort in mobile robot system and was attempted by the Stanford Research Institute. This contained two free wheels and two independent driven wheels propelled by stepping motors, a vidicon camera and an optical range finder along with several touch sensors. All elements were controlled by a single remote computer (PDP-11) which was radio linked to the onboard instrumentation.

(ii)JASON - This was developed by the University of Berkeley between 1970 and 1975. This had a front free wheel and two rear wheels driven by DC motors with ultrasonic range finders. A computer (PDP-11) was connected to an on board processor (8Ø85).

(iii) JPL "ROVER" and the CMU "ROVER" developed in the early 7Ø's by the Jet Propulsion Laboratory and Carnegie-Mellon University. They were similar to 'Jason' mentioned above.

Other models like the ANU (Australia National University) mobile robot, Yamobiko (University of Tokyo) MELDEIC, MELDOG, CART and VESA are amongst the recently developed mobile robot systems. Out of these VESA I and II developed between 1983-1985, are the only semi-autonomous robot with local environment perception and decision making capabilities.

The Teleoperated Vehicle developed here earlier consists of three driven wheels mounted on a shaft coupled to two stepping motors each of 196 N.cm. torque capacity. The two motors are used to control the motion of the vehicle. The position of the vehicle is represented and monitored on a $(r,\theta)$ coordinate system where one motor controls 'r', i.e., the radial movement while the other motor control '$\theta$', i.e., the angular direction of the wheel shaft (fig(1a)). Thus any point on a two dimensional plane can be accessed with proper movements of the two motors. The transmission of power from one of the motors is through belts while from the other the power is transmitted through bevel gears (fig (2)). The stepper motors are controlled by a controller which is capable of accepting external clock and direction signals. The controller is connected to a microprocessor (8Ø85) and the movement of the



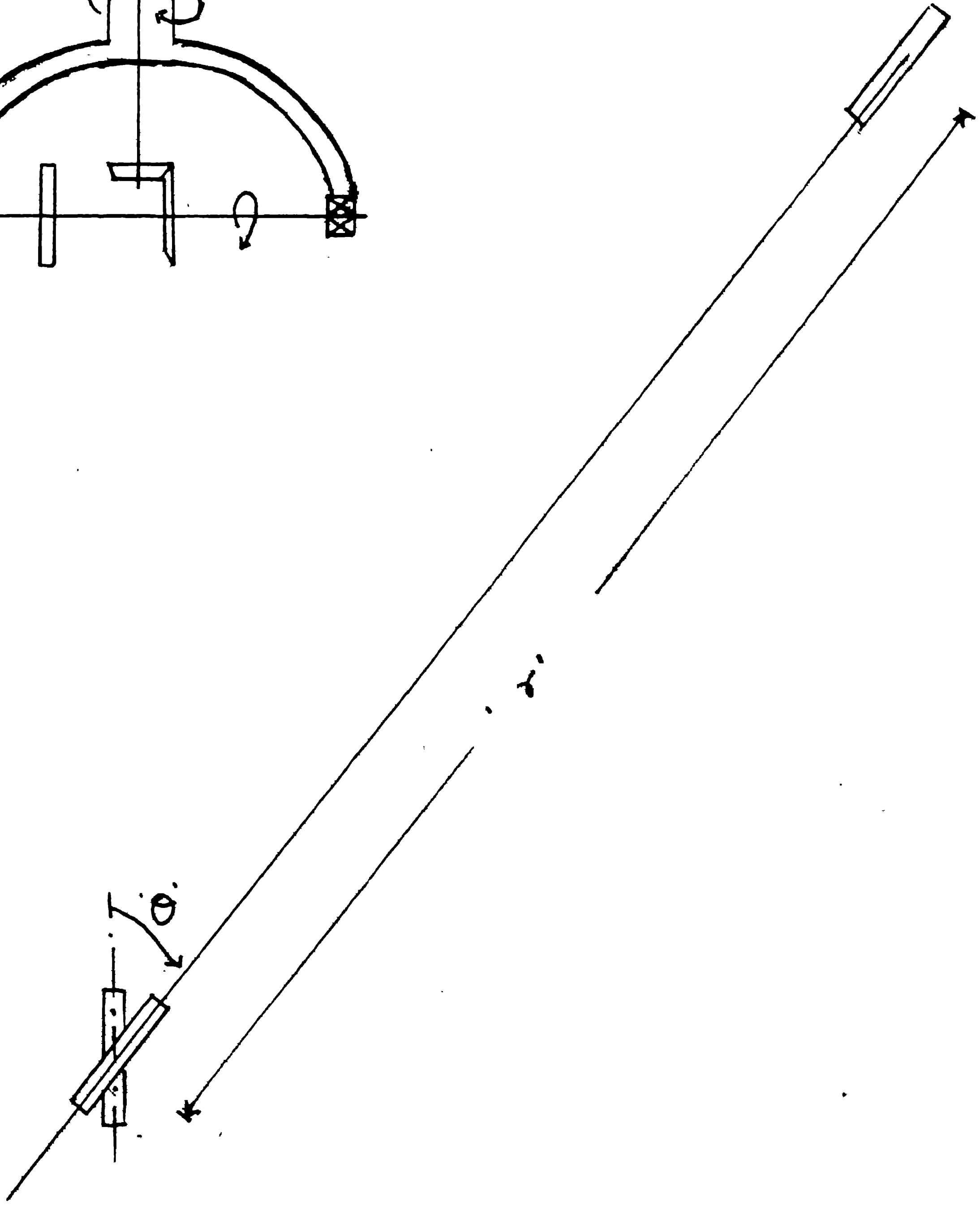


Fig. 1

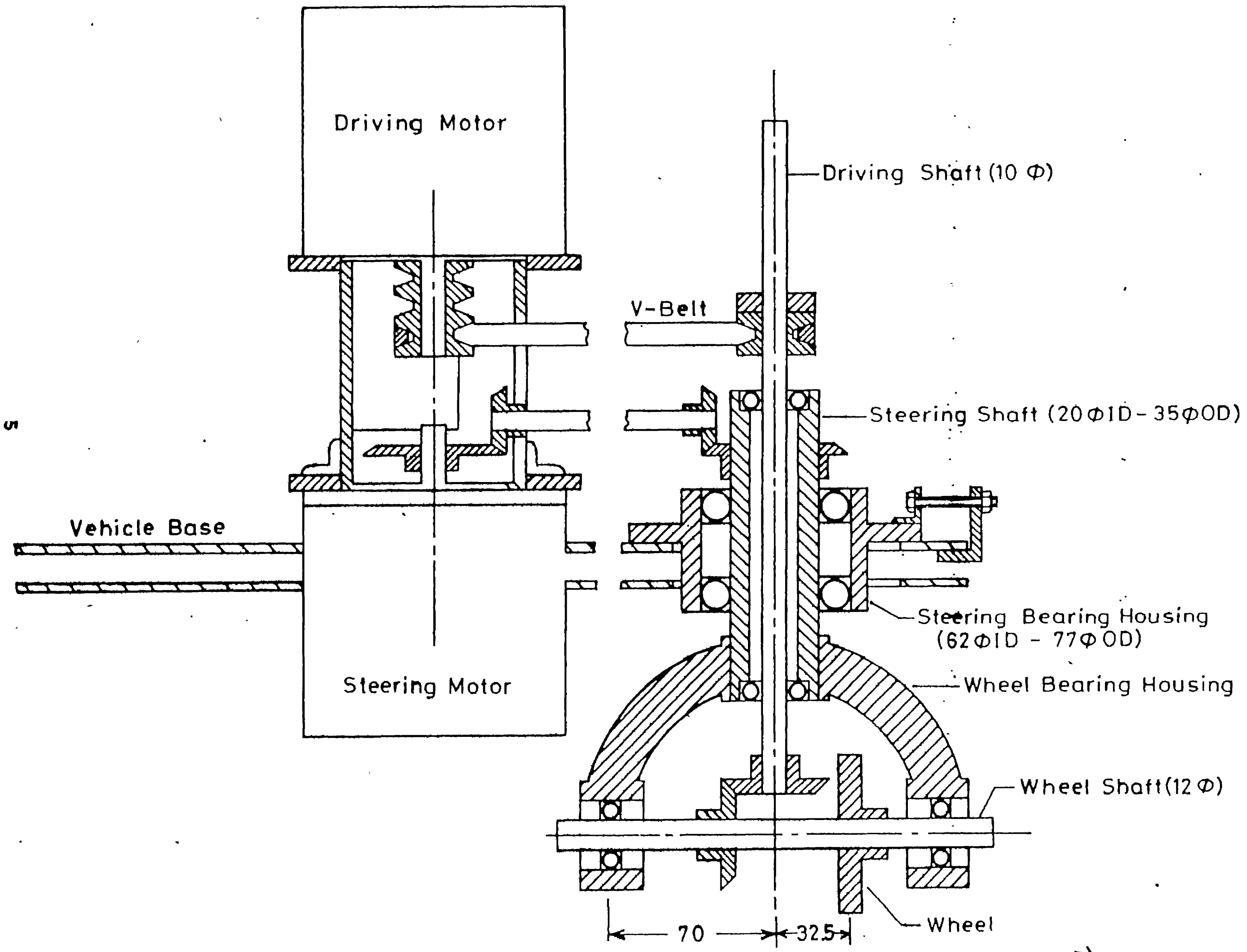


Fig. 2 Assembly drawing for one wheel. (ref. [3])

vehicle can be controlled so that the required path is traced by means of the number of steps needed to traverse in a given direction. The data for the number of steps needed to move the two motors have to be fed manually and hence complete automation flexibility cannot be acheived. Any path that has to be traced has to be manually fitted beforehand and the exact number of steps required to move the motors are to be computed. This was rather cumbersome and some data are bound to be lost during operation [3].

## 1.3 OBJECTIVE OF THE PRESENT WORK:

The main objective of the present work is to make the path planning easier and to rectify certain mechanical defects of the system mentioned above. The flexibility of the system is enhanced by getting the path completely planned using a high level language. A Pascal (or any other high level) program executable on a P.C has been used to send some comprehensive data to the 8085 processor of the stepper motor controller . The provision of sensors makes the system more versatile. Once the path is planned the vehicle can start moving and the host processor is relieved to execute other programs for controlling the manipulator. In the event of any mistake in the planned path or on encountering any unforseen obstacles provision for replanning has to be included in order to increase the flexibility of the system. The two mechanical units (vehicle-Robot) has to function as one system. This calls for integrating the two units making them as one. The manipulator arm is to be mounted on a flat surface on the vehicle. Such a mechanical system however has limitation in its range and can reach only those areas circumscribed by the maximum length of the cable. This causes loose wires lying on the floor making the wheels suceptable for entanglement. This calls for the cable to be taut at any instance. The combined Vehicle-Robot unit can be used to perform the required tasks in two different work environments. The task may consist of picking an object from place 1 and keeping it at place 2 which is outside the normal work envelope had the robot been fixed

## 1.4 SCOPE OF THE PRESENT WORK:

The mobile robot system developed can be used to traverse in either of the two modes i.e.,

(1) by means of explicit programming where the user completely specifies the via points and the start and the end points

(2) by means of Model Programming where the user defines the geometry of points and a description of the task in terms of the models.

The range is limited by the length of the cable. However the problem of loose wires hanging all over the place (drawback mentioned in [3] ) is overcome by means of a simple wire winding/unwinding mechanism. The system can be simulated for proximity sensors whrerein the location of the obstacle can be retrieved and the control of the system is passed over to the human operator for replanning its path. The Visibility Method [4] is implemented for Model based path planning and cubic interpolation is used for the explicit programming part of the software

# CHAPTER - 2

# HARDWARE DESIGN

## 2.1 INTRODUCTION:

The hardware in this project can be divided into two headings namely (1) Meachanical Hardware and (2) Electrical Hardware. The mechanical suitability of the vehicle needs some reconsideration as is to be explained later. However the drawbacks of a three support system with regard to its stability warrants additional supports to stabilise the vehicle. This is provided in the form of Castor wheels, also known as trail wheels (two in all). The mechanical hardware redesign rectifies the defects in the transmission system. The electrical hardware establishes a two way communication between two 8Ø85 based microprocessor forming a part of the stepper motor controller and the SIR-1's (the manipulator to be fixed on the vehicle) controller and a 8Ø286 based P.C. (H.C.L., Busy Bee). The RS 232A protocol is used for this purpose. An extension card containing a 8251A USART is also added to the hardware since the 8Ø85 kit is devoid of it.

## 2.2 DESIGN OF THE VARIOUS MECHANICAL COMPONENTS:

The design of the various sub systems are explained in following sections.

### 2.2.1 TRANSMISSION SYSTEM:

As has already been mentioned earlier [3],the vehicle developed had power transmitted from one of its two motors with V-belts (fig(2)) while the second motor transmitted power through a bevel gear arrangement. The belt transmission system is not a positive drive and hence on application of excessive loads slippage takes place. Moreover, the poor surface contact between the belts and the pulley resulted in ineffecient energy transmission. The effeciency of the transmission can be improved by incorporating a positive drive system in the form of gears. The design and suitability of the gears had to be based on

(1) Ease of Manufacture,

(2) Transmission Effeciency,

(3) Suitability in the System.

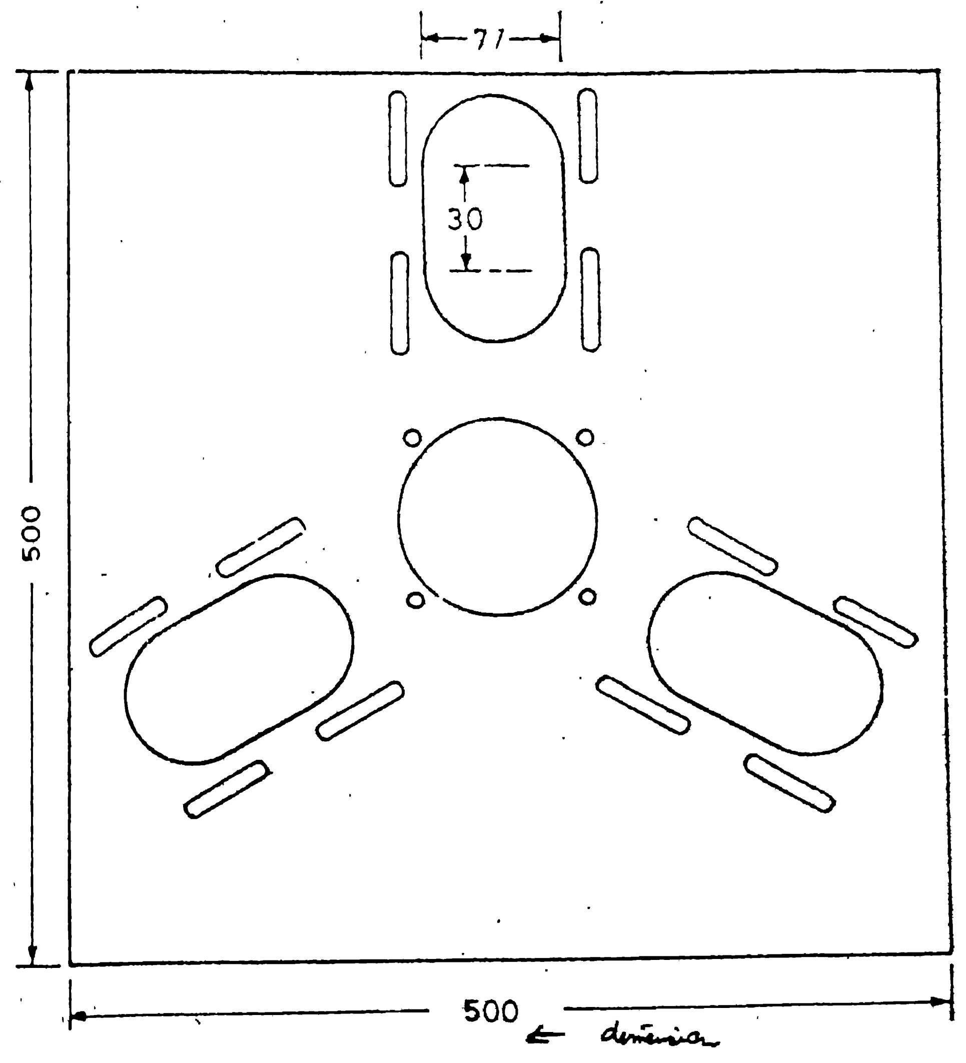
The axis of the driven gears form the vertice of an equilateral triangle and the axis of the driver(motor axis) is located at the centroid of the triangle. Spur gears were selected to transmit power to the three wheels. However the rather large center to center distance between the driver and the driven wheel axes, warrants the use of a light material for the gears. The light material should be strong enough to transmit the power from the driver shaft to the wheel. Further, the high inertia of such large gears could prohibits their use directly. The gears made of bakelite , further made light to take care of inertia by drilling 8 equally spaced holes ($45^{o}$ from each other) are used for this purpose. The dimension of the gears after the usual check for the stresse [5,6 ] are listed below :

| | Gear | Pinion |
|---|---|---|
| Blank Diameter | 182 mm. | 175 mm. |
| Module | 1.Ø5 | 1.Ø5 |
| Pitch Circle Diameter | 179.9 mm. | 172.9 |

|  |  |  |
|---|---|---|
| Width of the Blank | 1Ø mm. | 1Ø mm. |
| Thickness at P.C.D | 2 mm. | 2 mm. |
| Number of teeth | 171 | 165 |
| Depth of teeth | 2.2 mm. | 2.2 mm. |

## 2.2 CASTOR WHEEL FIXTURES:

According to the principle of location, three and only three pins (points) are needed to locate a plane [7,8]. The three wheels for a vehicle provides the required location in a plane . Since we are considering a plane parallel to the floor, 3 fixed drivers are sufficient. All the same the three support system is not mechanically stable (as in the case of a tripod) and the system tends to topple. Additional supports have to be provided to overcome this drawback. This is done by using castor or trailing wheels. The castor distance or 'trail' plays an important role in deciding the torque needed for the castor action [9]. The castor wheels when mounted should provide the support needed without in any way affecting the advantage achieved due to its three support. The castor wheel mounting should, hence be telescopic, permitting free up and down motion to keep the vehicle parallel to the plane below. The restoring torque for the castor wheel with a trail of about 1Ø mm. ranges between Ø Nm. to about 5 Nm. conisidering the two extreme positions of the castor wheel (fig.(3)). On considering the work envelope of the manipulator arm one can easily conclude that the two castor wheels mounted in the front (fig.(3a) - fig.(3b)) would suffice to overcome toppling.



All Dimensions in mm.

Fig. 2 (b) Vehicle base.

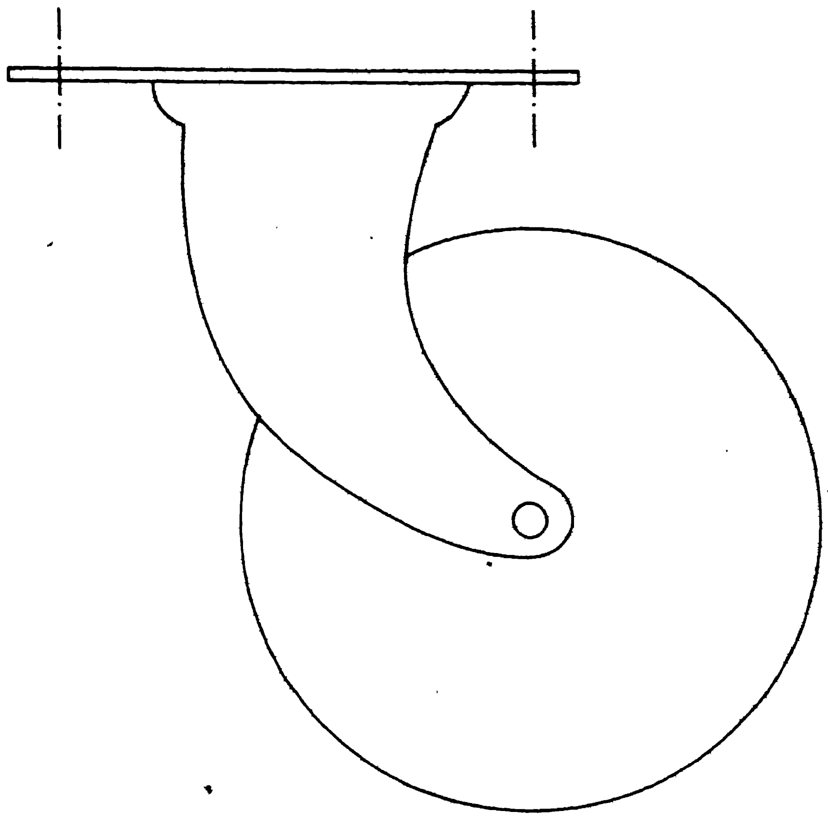
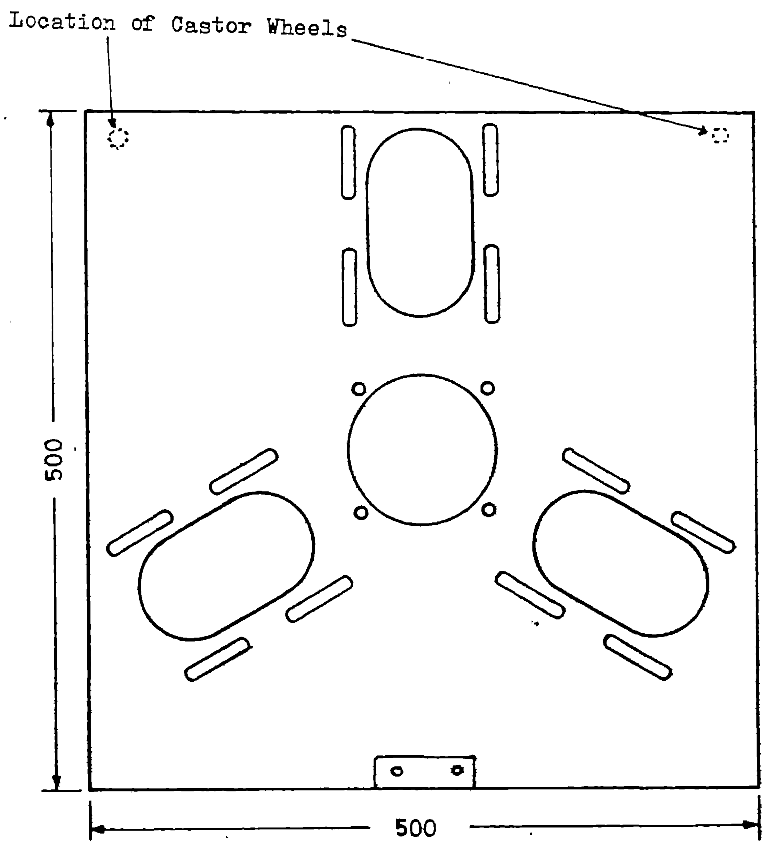


FIG. 3. CASTOR WHEEL



All Dimensions in mm.

Fig. 3 (a) Vehicle base (Modified)

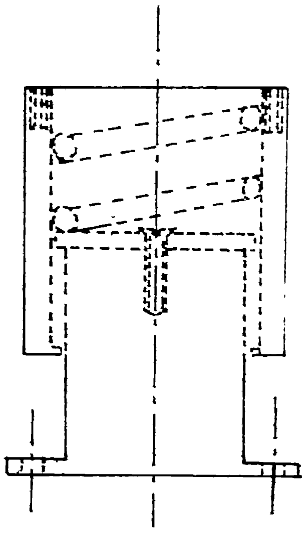


Fig. 3 (b) Castor Wheel Fixture

### 2.2.3 MOTOR HOUSING:

The housing for one of the motors had to changed completely because of the change in drive while that of the other motor had to be altered slightly. The new housing for the new transmission has to serve the dual purpose of (1) providing the housing and (2) provide a proper base for the manipulator arm to be mounted on it. The housing also has to have slots cut so as to allow the meshing of the gears. The details of the housing (made of aluminium)is showed in fig.(4) The robot manipulator is mounted on the flat surface atop the housing through another flat plate.

### 2.2.4 OTHER MODIFICATIONS:

Over and above the above major modifications, a few minor modifications were brought about. The shaft coupled to the bevel gears was earlier housed on a cast iron housing and was fixed to the aluminium body of the vehicle. But due to the low shear strength of aluminium the threads could be easily be sheared off. Thus the housing could not be fixed as firmly as is required. This problem is overcome by placing mild steel flats of about 4 mm thickness with tapped holes between the two aluminium plates of the body (fig.(5)). Since mild steel has greater shear strength than aluminium, the housing can be secured firmly. Moreover, by placing a sheet between the plates the entire assembly can be easily aligned for proper meshing by letting the gears to stabilize when the housing is not fixed (and allowing power transmission to take place). To improve the load

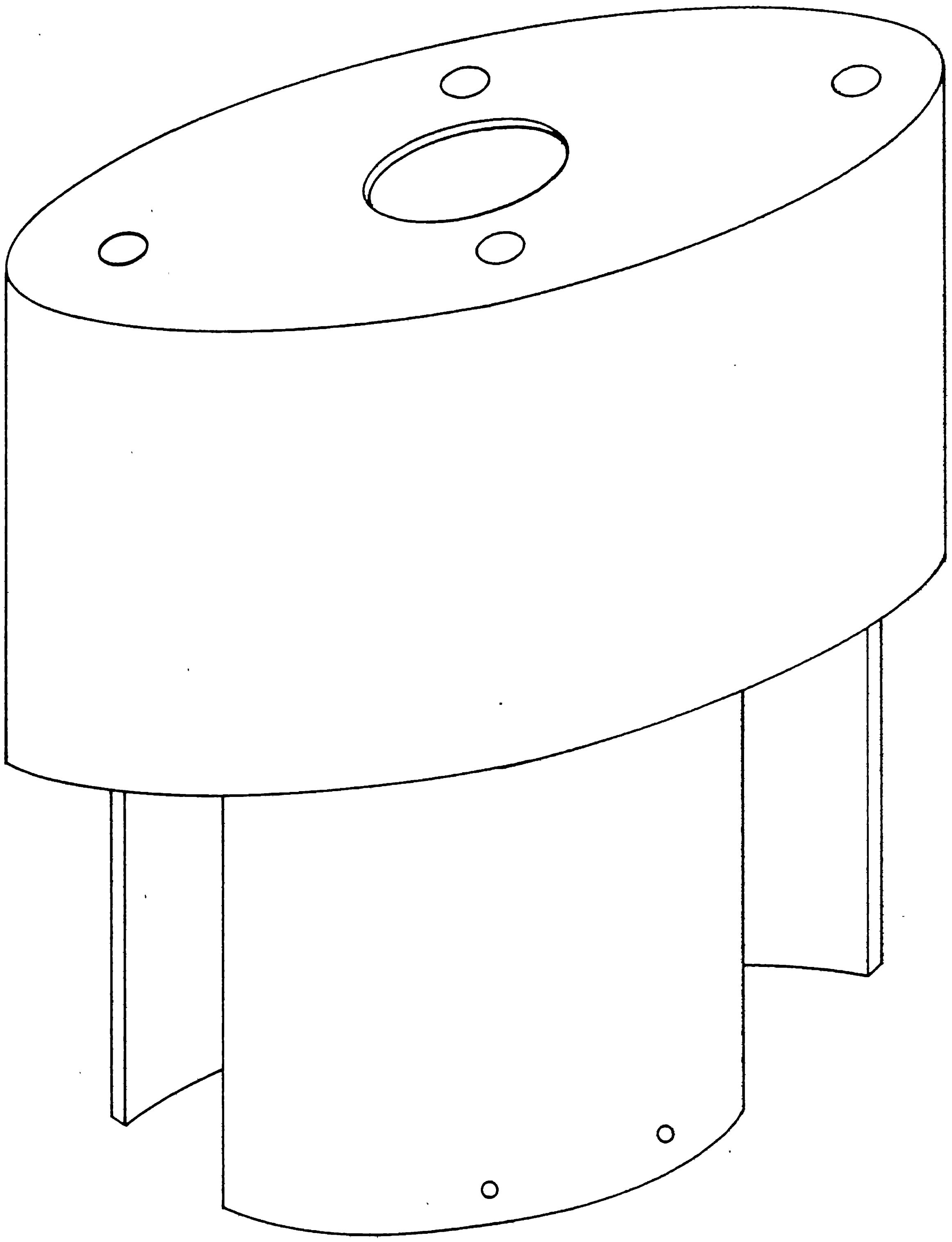


FIG. 4. MOTOR HOUSING.

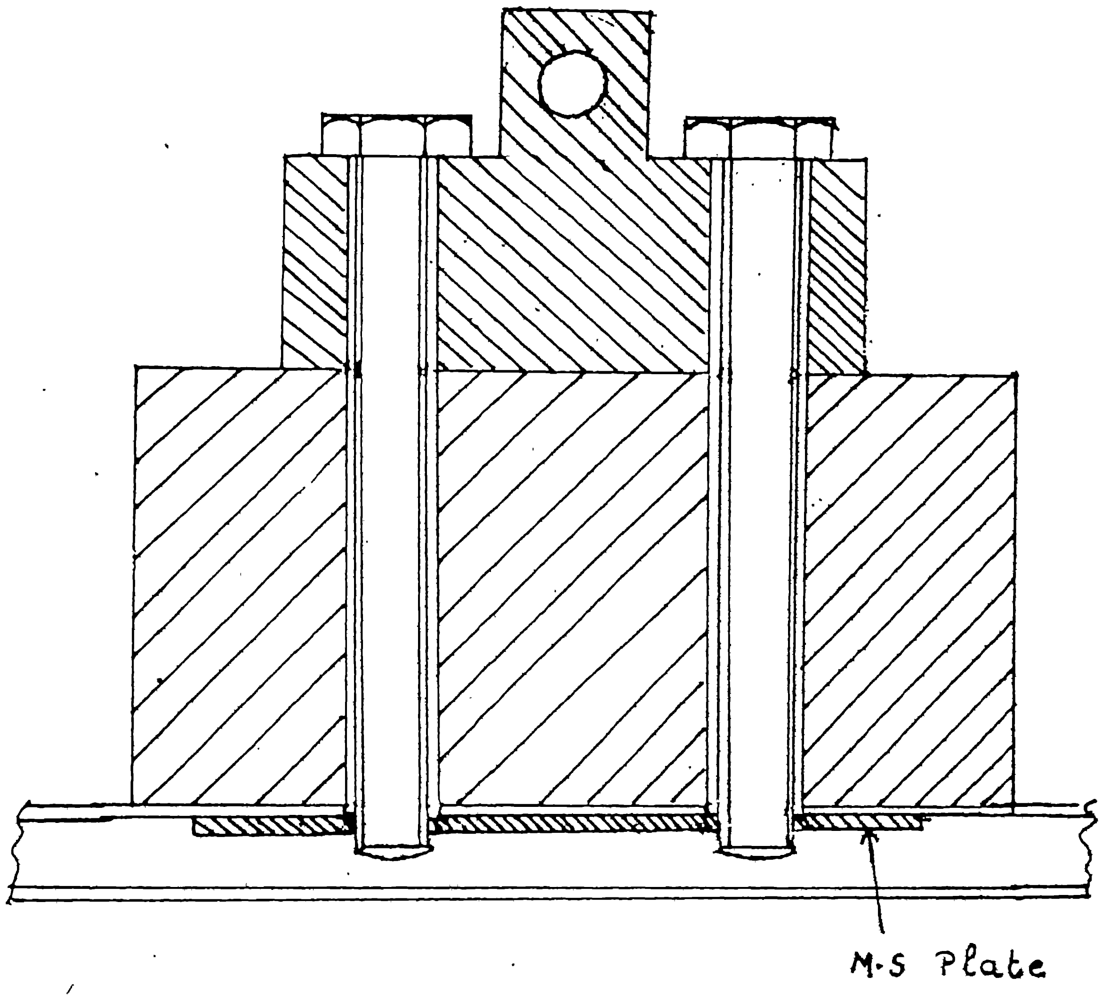


Fig. 5

carrying capacity of the body without buckling, more reinforcements in the form of perspex square (1/2") are placed.

## 2.3 WIRE WINDING/UNWINDING MECHANISM:

The range of the vehicle as mentioned earlier is restricted because of the length of the cable used. Moreover by having long leads the cable tend to lie all over the floor space making the wheels succeptible to entanglement. Ideally a wireless communication system should be used for carrying the signals. The alternative would be to have just the sufficient amount of cable between the control station and the 'MOBILE ROBOT'. Such a mechanism should be in a position to sense the tension of the cable to decide whether there is excess wire lying on the floor or not. The mechanism comprises of a large drum over which the cable leading from the controller to the vehicle is wound, a pair of guide rollers and a sensor roller. The drum is connected to a low speed motor. The cable is led through the guide rollers and the sensor roller where the cable tension is sensed. The details of the various mechanical parts used is discussed briefly below.

### 2.3.1 DRUM:

The drum used must be large enough to restrict the amount number of rotation of the drum for a given length of the cable. Since the cables are fixed at both ends (although one of its ends can move, for our purpose we consider it fixed since it is not free here), the wire has to be brought out from the center of the shaft supporting the drum. The drum is connected to a small motor through a belt. The drum should contain flanges on shaft is supported on bearings to permit free rotation of the

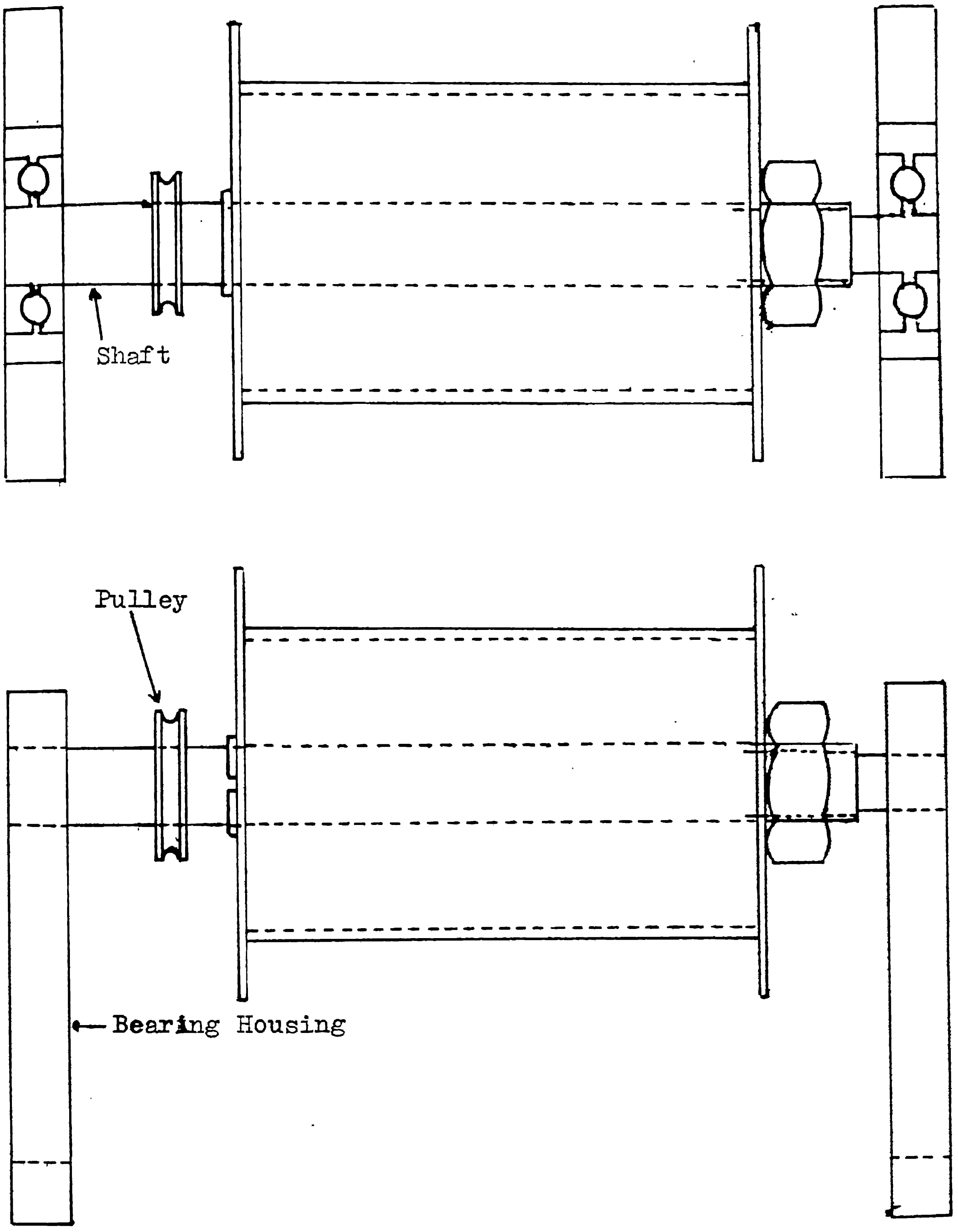


Fig 6 Drum

drum. (fig. 6)

## 2.3.2 SUPPORT ROLLERS:

The support rollers should be in a position to guide the cable freely and provide sufficient contact area for the sensor roller. The rollers are used only for guiding the cable properly (fig.(7)).

## 2.3.3 SENSOR ROLLER:

This is the most important part of the mechanism and is used for sensing the tension of the cable passing over it. The ends of the shaft of this roller is housed on a square block capable of sliding in a guide. The square block is supported on soft springs. The movement of the springs causes the making or breaking of an electrical circuit which powers the motor attatched to the drum.(fig(8)).

The cables are wound over the drum such that sufficient wire is there to allow the free movement of the robot (this mechanism is connected only to the cable of the vehicle). The cable is then passed below one of the rollers and then allowed to pass over the sensor roller before finally passing below the second support roller for use at the vehicle end. The cable when not pulled causes the spring to return to the null position wherein the circuit is made due to contact of two brass rods. On application of some pull on the cable at its moving end (vehicle end) the sensor roller moves down slightly along the guide and the contact is broken.(fig.(9)).In normal use this mechanism should be operated if and only if the motor controlling the radial direction is on (i.e., if the motor controlling the

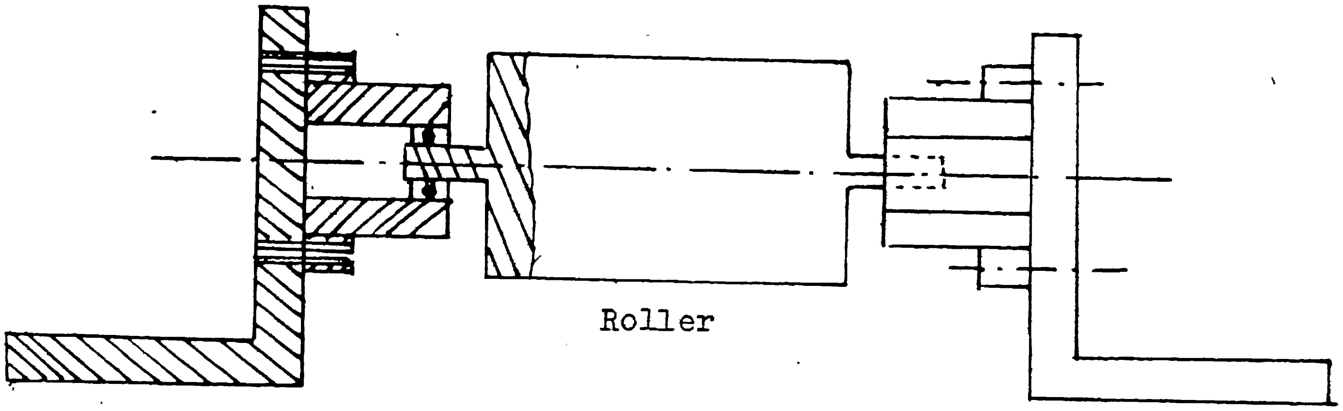


Fig. 8 Support Roller

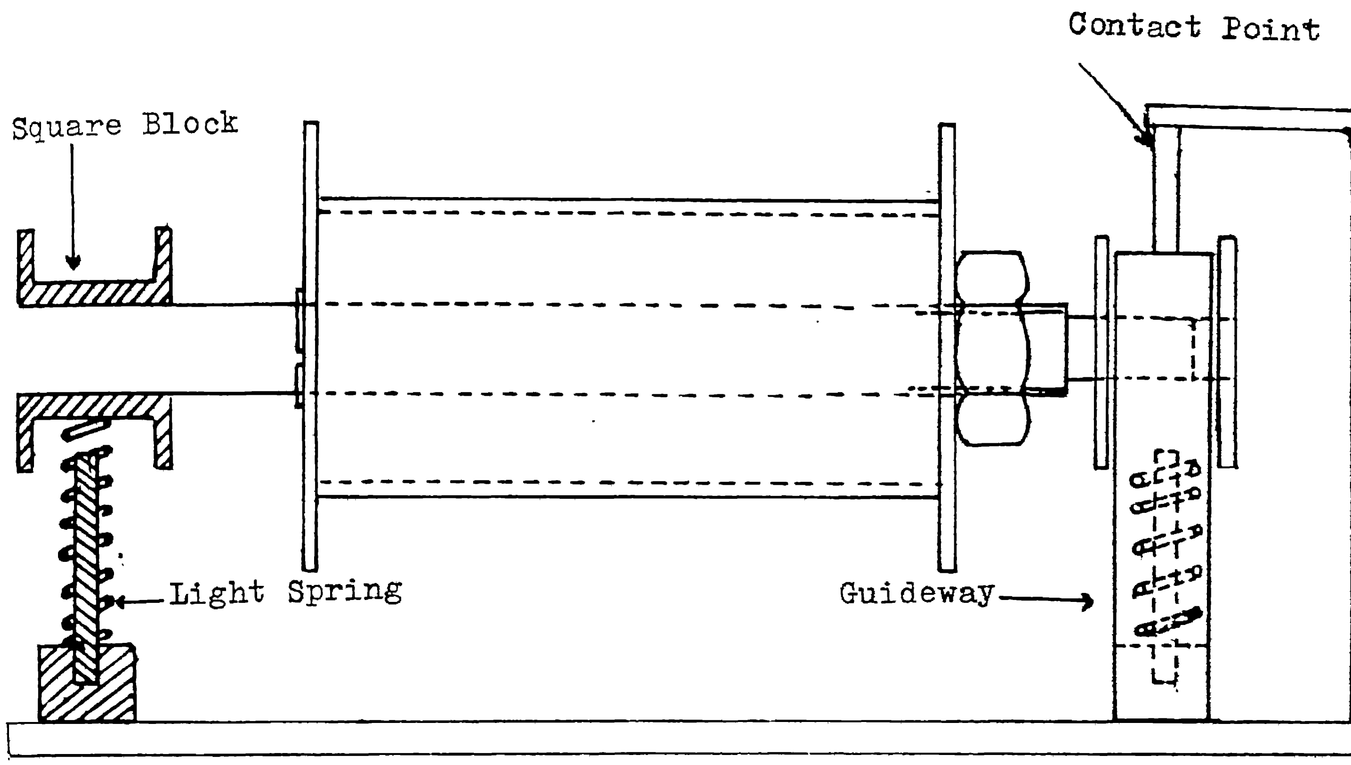
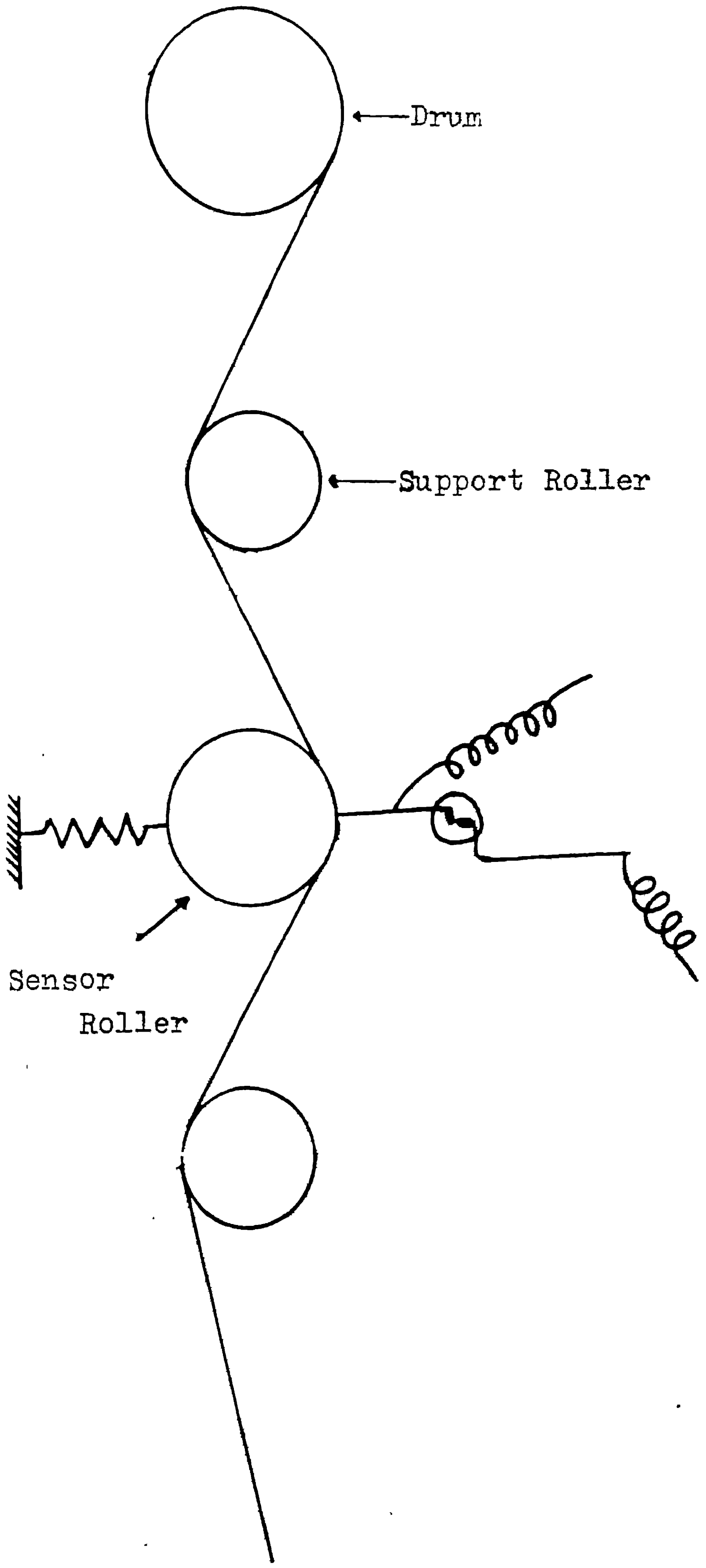


Fig. 7 Sensor Roller


Drum
Support Roller
Sensor Roller

spur gears rotates).

## 2.4 ELECTRICAL HARDWARE:

The utility of any systen depends on how easily the user can handle the same. The exhaustive mathematical calculations requires a user-friendly software. The machine language operation of the microprocessor is difficult to manage and hence the complicated computation has to be done else-where on a computer. Thus having decided to perform the path planning(the number of steps to be moved by the two motors [3] as mentioned earlier) on a personal computer, a suitable method of transferring the data from the memory of one system to that of the other is needed. Of the various available methods a serial data transfer is preferred. The 8Ø85 kit is devoid of the serial interface chip and hence an extension P.C board with 8251A chip and the supporting line driver are used to interface the P.C and the uP. An address decoder in the form of 74 LS 138 is used to address the 8251. The address of the chip is FF and that of its register is FE. The support chips includes 74Ø4 (inverter), 7486(XOR gate),7416Ø(decade counter),74161(binary counter). The signals are made available at the end of a standard 25 pin D-type connector. The baud rate (the number of bits sent out of the communication port/second) of 96ØØbps is used for data transmission fig(1Ø). The other electrical hardware include an INT 6.5 connection for simulating an interrupt in the path of the 'MOBILE ROBOT' due to some unforseen obstacles comming in its way. The wire winding/unwinding mechanism also needs supply for operating the fractional horsepower A.C. induction motor. The

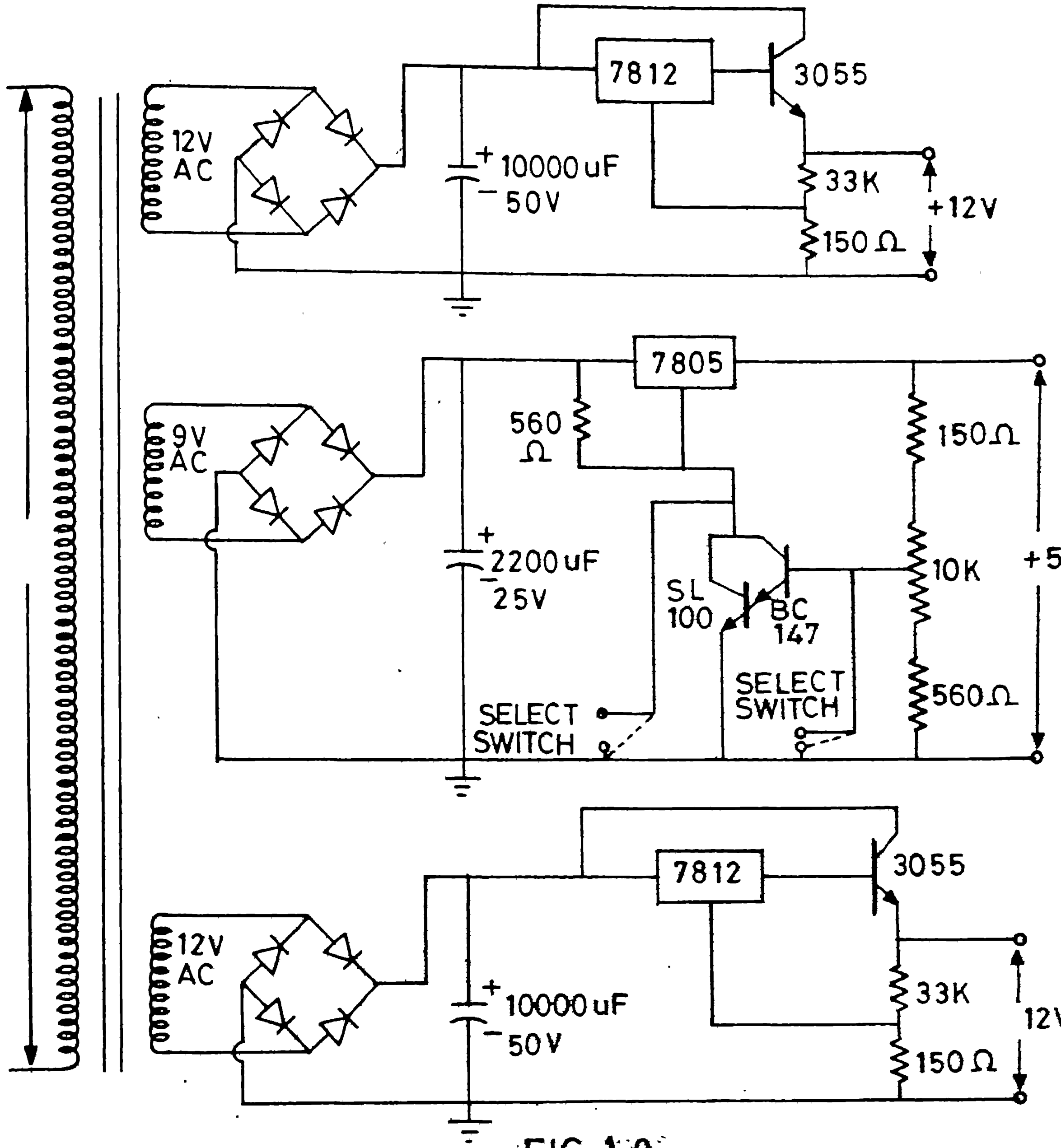


FIG.10

Ckt diagram for power supply (ref. [3]

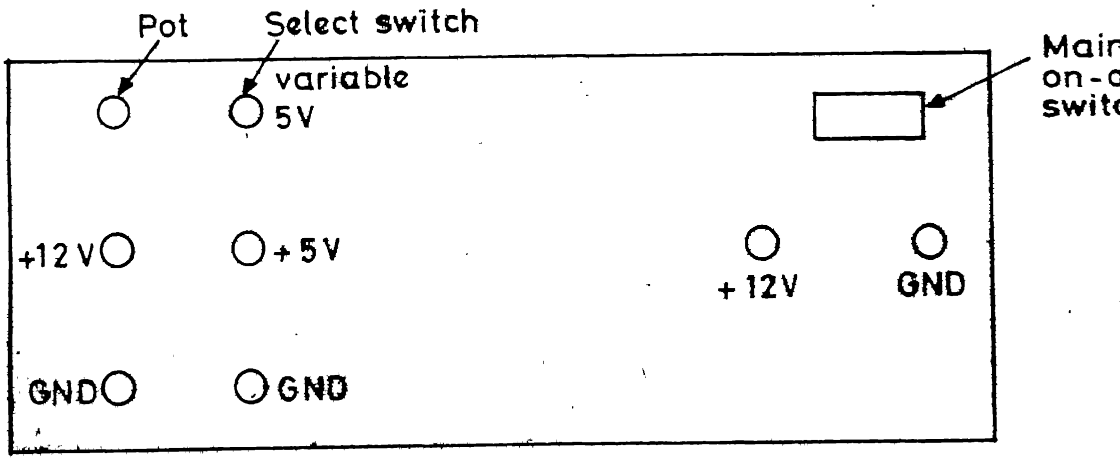


FIG. 11

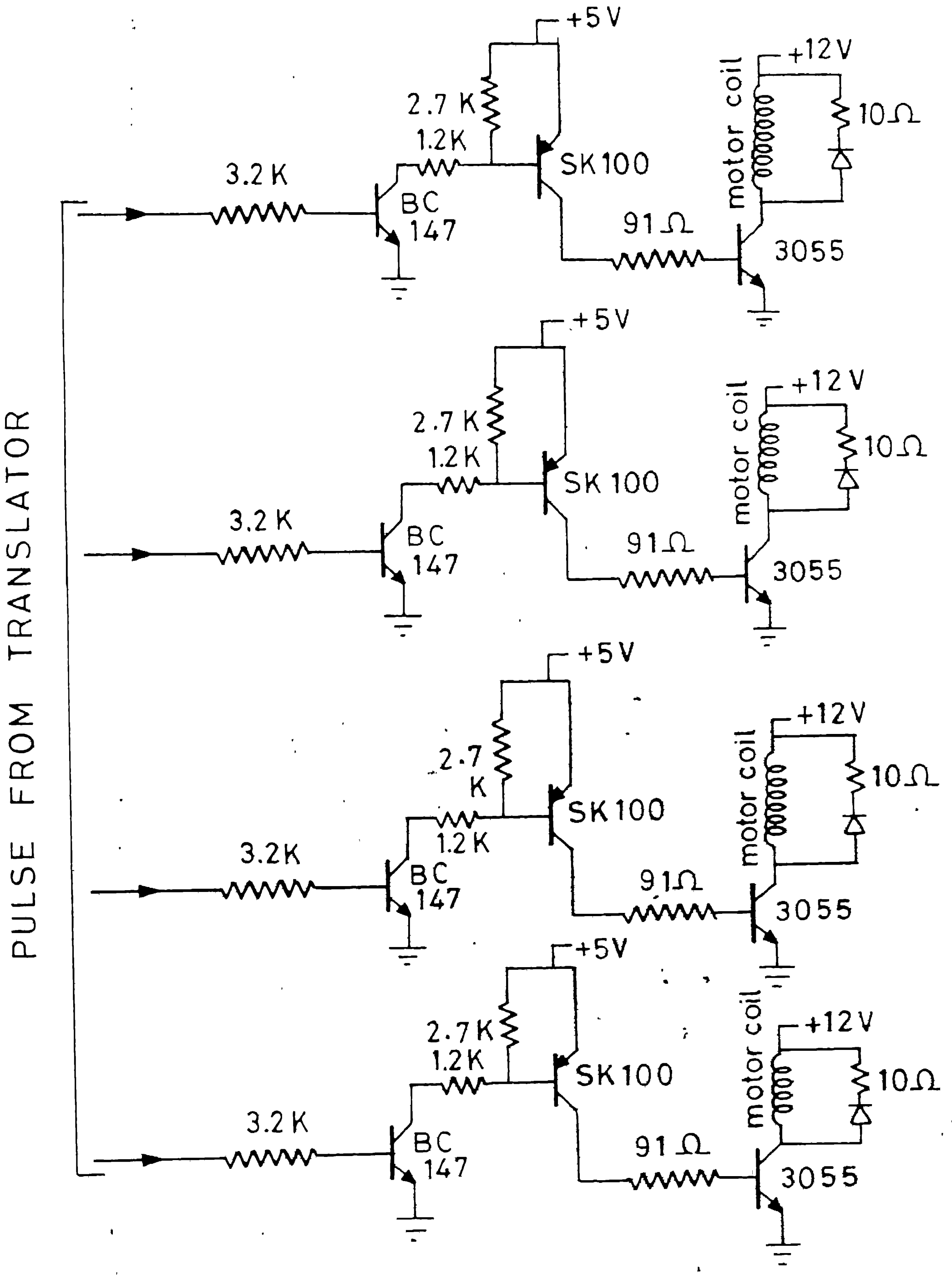


FIG. 12

DRIVE CIRCUIT FOR THE MOTOR (ref. [3])

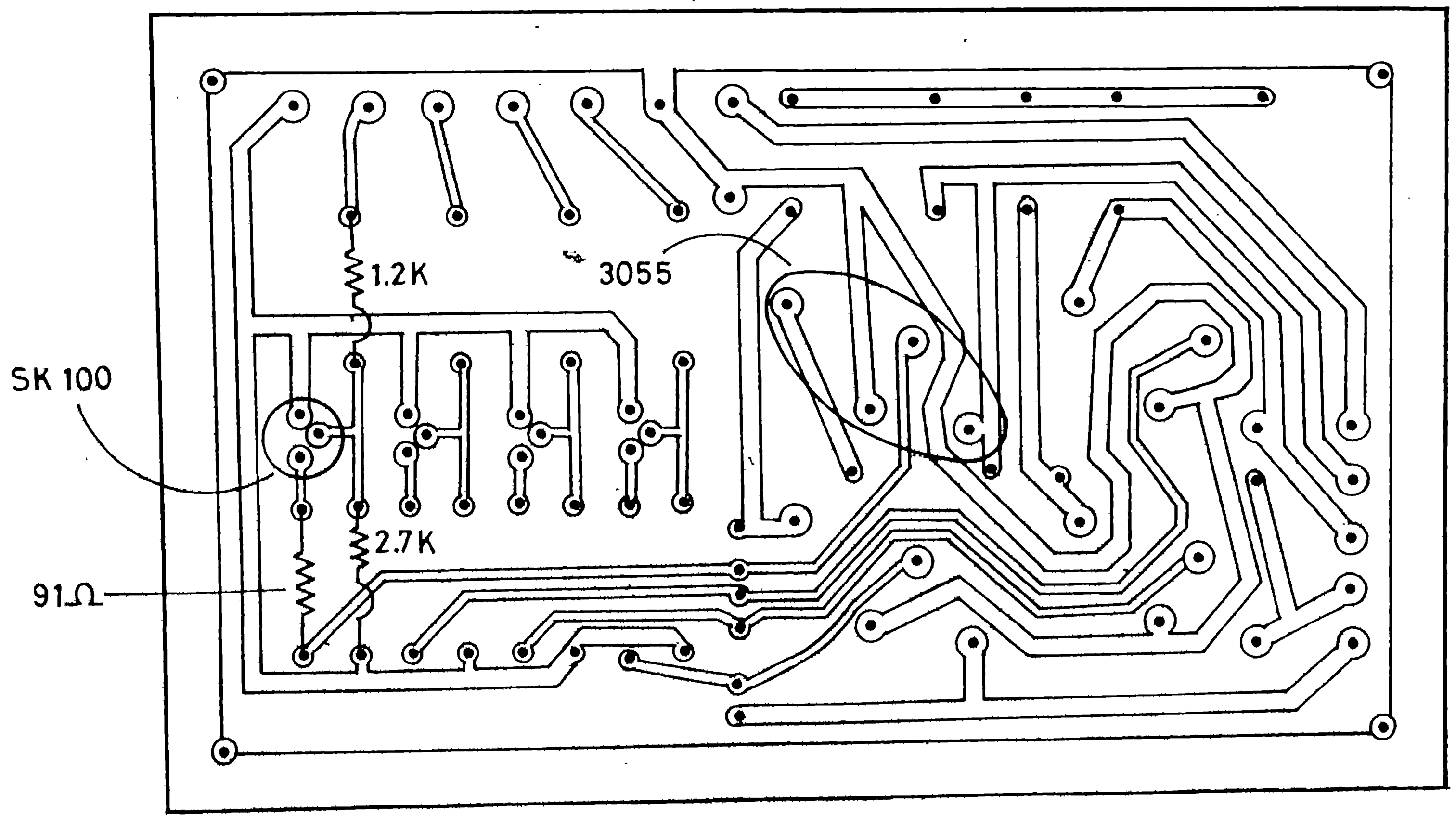


FIG.13 PCB LAYOUT FOR DRIVE CIRCUIT (ref. [3])

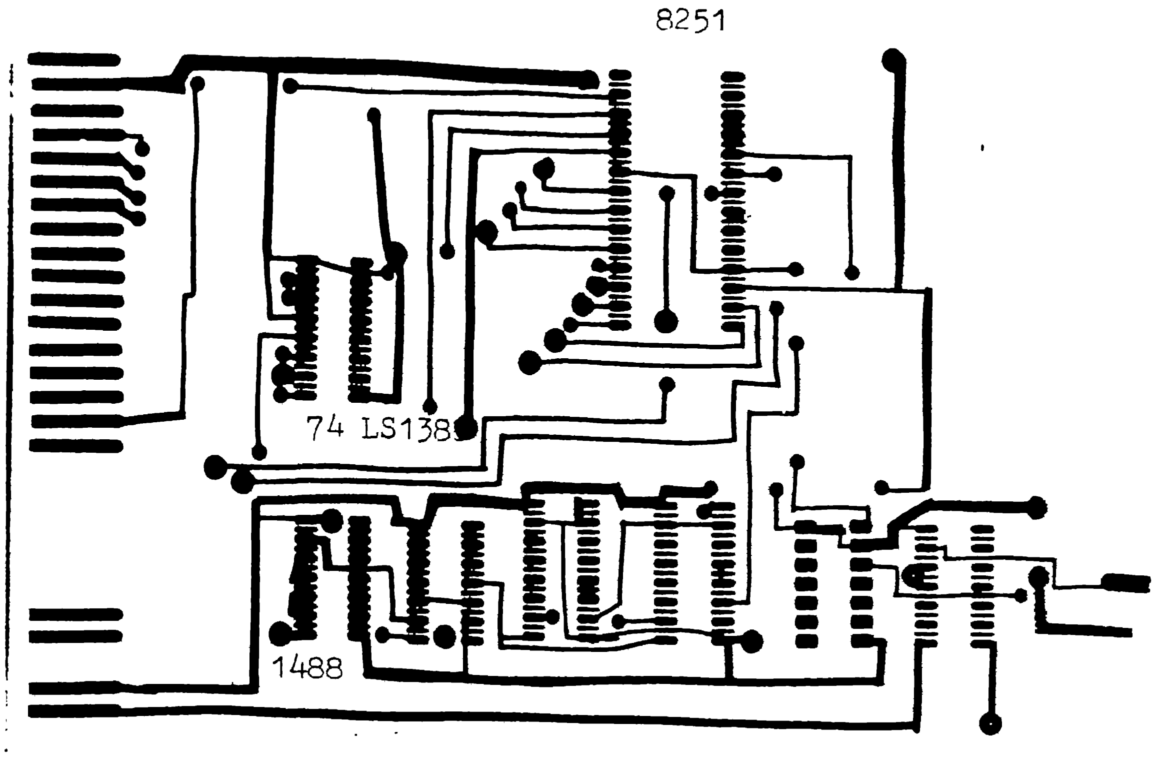


Fig. 14a : PCB Layout for 8251 (Bottom)

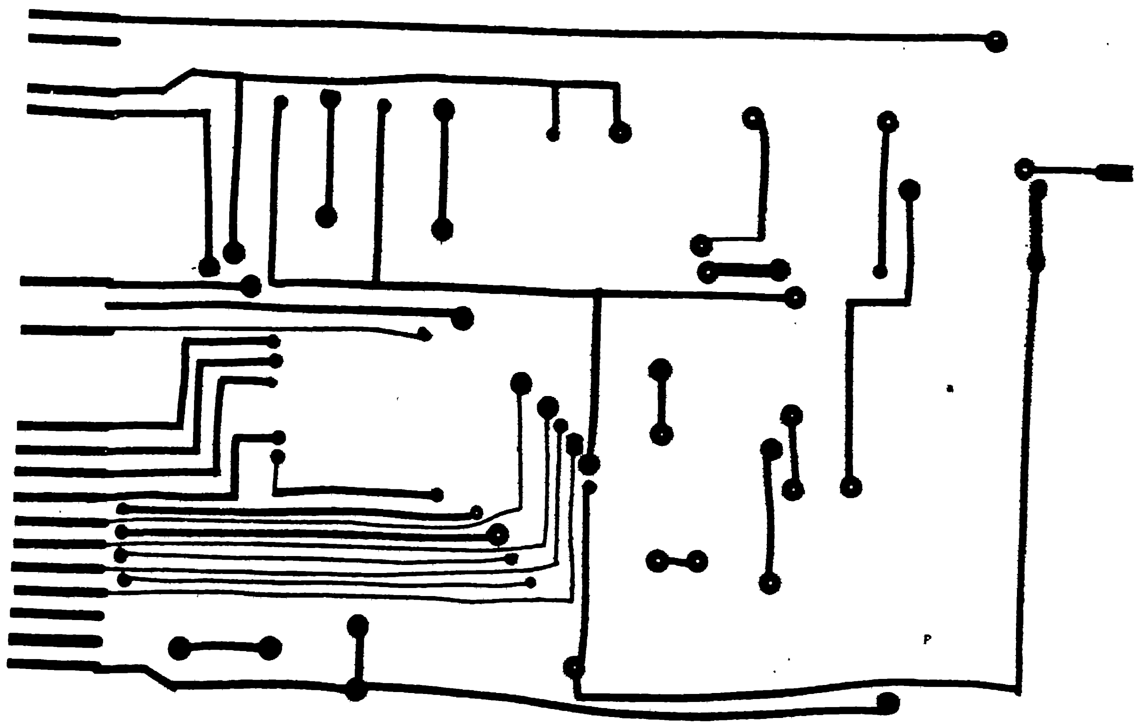
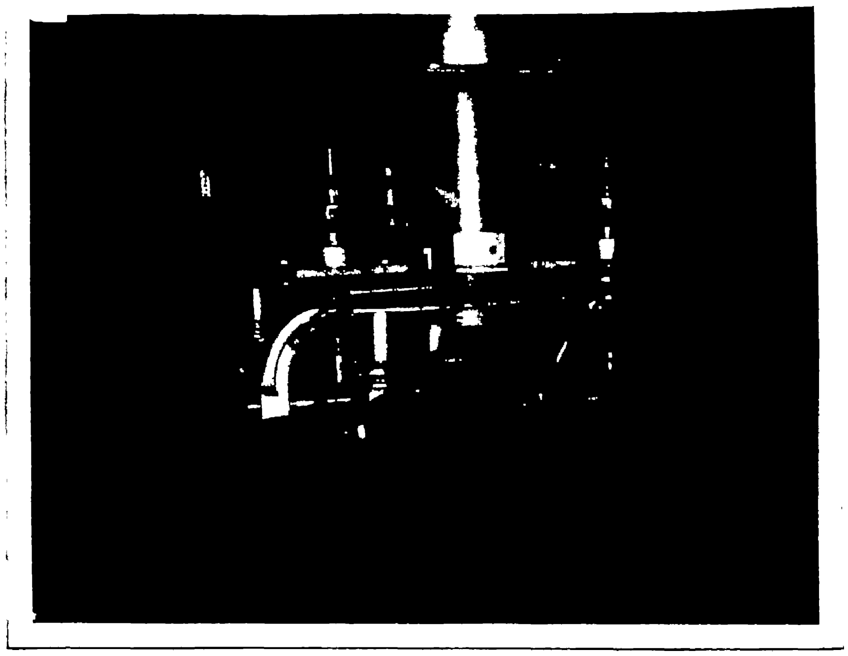
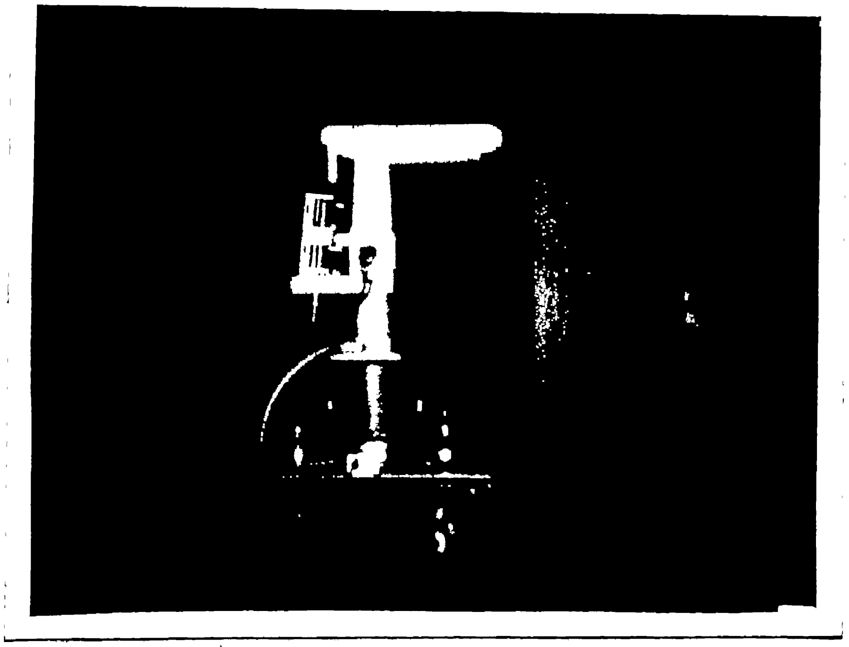
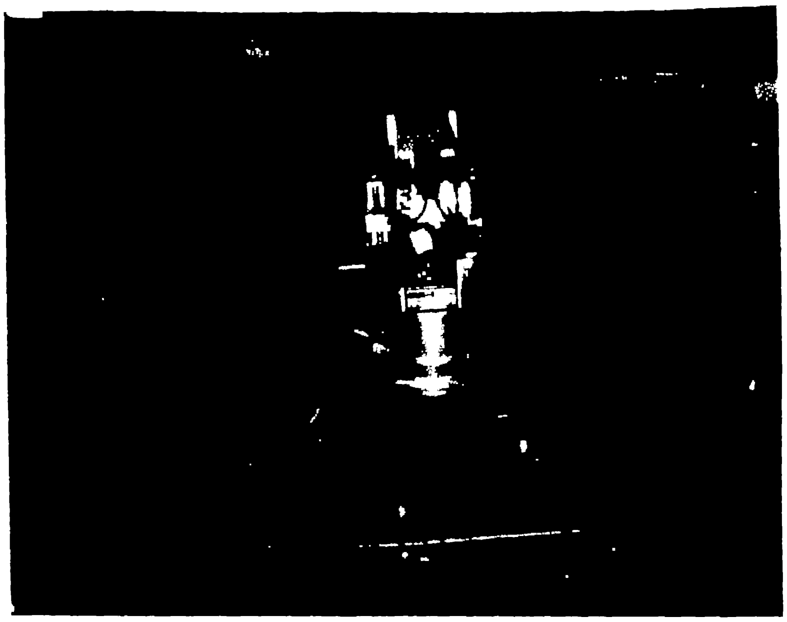


Fig. 14b : PCB Layout for 8251 (Top)

supply for this motor should be given only when the vehicle translates.



# CHAPTER - 3
# SYSTEM SOFTWARE

## 3.1 INTRODUCTION

The present setup, as has been indicated earlier is intended to contain a host processor which can independently control the working of the two mechanical subsystems : (i) the Carrier and (ii) the Robot. Since the controller of the carrier (Vehicle) and that of the manipulator contain seperate microprocessors,the software has to be implemented in both the host P.C. (with 8Ø286 processor) and the system each with 8Ø85. The path planning of both the vehicle and the manipulator is done in a macroscopic order in the P.C. and in a microscopic order in the system processors. The flexibility of the system depends directly on how easily one can use it. Hence the software has to take care of the workspace to decide the type of approach to be used.

The simplest trajectory planning for a manipulator system is usually done by giving a set of via points (if needed), the start and the end points to be connected in a suitable manner. With the points known and with a few assumptions, a piecewise cubic curve can be fitted considering three points at a time [9]. This type of programming also known as explicit programming is not always suited to the shop floor problem in which the manipulator is usually expected to work. Model programming however varies from explicit programming in that the user specifies a set of geometric models of points and a description of the task in terms of the models. The manipulator (vehicle- robot assembly) motions are derived by the assembly system from these specifications. Several model based algorithms are available and all of them are in their experimental stage and under development. Udupa[13], first formulated the obstacle avoidance problem by considering the moving object as a point. We are concerned with fixed orientation of the moving object thus making the problem easy. Many other researchers have also studied this problem and suggested the possible solution for the same [4,1Ø,11]. The shortest safe path between two points is the peicewise linear paths connecting the start and the goal through the vertices of the various polygons . The vertices are selected such that the line joining any two successive points do not intersect with any polygon. Thus the selected path would tend to go along the sides of the polygon or would join directly two points in two different polygons. Normally, we are more interested to reach the destination avoiding obstacles if they come in its way and at the same time selecting an optimum path between the start and the end. The second part, i.e., the optimization of the path, has numerous solutions called differently by different people and is typically referred to as "Critical Path Method" problem.

The first part is a state-space problem [12]. A start node is associated with the initial state description and successive nodes are calculated using operators that are applicable to the state description associated with the node. Many methods are suggested for avoiding obstacles. Intuitatively speaking, one can reach the goal if one can see it from some other station and continue proceeding down the tree till the start station is reached. Such a process, known as the "Visibility Method" suggested by Lozano-Perez[4], is implemented for the Obstacle Avoidance algorithm here.

### 3.2.1 VISIBILITY METHOD:

As already mentioned, this method provides a continuity graph and matrix between the various corners of the obstacles present along with the start and the end. The continuity graph is searched to obtain the required end state [12] . This graph is obtained by checking whether the line joining any two state points intersect the diagonals of any of the obstacles. If the visibility ( direct connection ) is there then the matrix element is entered with the distance between the two points in question. The same can be checked using sides of the polygon instead of the diagonals. Here, however, the degree of computation is twice that of the former. The algorithm is implemented by using parametric notations for representing a line as below

$$X = X1+(X2-X1)*L1 \quad ....(i) \quad ; \quad Y = Y1+(Y2-Y1)*L1 \quad .... (ii)$$

$$X = Xi+(Xj-Xi)*L2 \quad ....(iii); \quad Y = Yi+(Yj-Yi)*L2; \quad .... (iv)$$

where X and Y are the X & Y cordinates of the point of intersection of the two lines represented by {(i),(ii)} and {(iii),(iv)} above, (X1,X2) & (Y1,Y2) are the X & Y coordenates of the start and the end points. If $0 \leq L1 \leq 1$ and $0 \leq L2 \leq 1$, then the two lines intersect somewhere between the start and the end location of the line.

3.2.2 SHORT PATH SEARCH:

Searching algorithms are widely classified as (i) Breadth first and (ii) Depth first search. The search in our case has to maintain continuity between two successive points and optimise the distance between two points. Mathamatically speaking, the shortest path is either the direct path or has to be along the straight edges of the obstacles [12]. The Dijkastra's Algorithm [13] has been used to identify the short path. This algorithm gives the critical path and has an order of computation of $(N^2)$(see appendix - 3). This algorithm is a breadth first search and is considered to be the most efficient short path algorithm available. The algorithm assigns a high index for all station except the start (initial pivot) which has an index Ø. Then at every stage, the minimum of the already present index is added to the distance between the pivot and the point in question. The minimum of all such values is identified to represent the node for the new pivot. This is carried on till the last point is reached. The connectivity is obtained by tracing the predecessor to each of the points used as pivots.

3.2.3 MICROPROCESSOR ALGORITHM (8Ø85):

The computed steps for the two motors have to be transferred to the microprocessor in order to trace a given path. The algorithm in the system should hence be made in a way so as to accept data and wait for the end of the data flow in order to perform the next task. The program initially sets the 8251A for a baud rate of 96ØØ and assigns a given memory location as a memory stack pointer. The data as received form the P.C are stored in a systematic manner. The end of data flow signified by a flow of 4 sets of zeros indicate the change of control to the stepper motor controller program. In the stepper motor controller program, the software delay is called every time an increment in step is needed. There is a count down of the number of steps already executed and the process repeats till all the given instructions are carried out completed. The location retrival takes place when there is an interrupt in the controller program. Here RST 6.5 is set when there is an external signal to the processor. The path planned by the P.C is divided into a number of straight lines, each straight line being furthur subdivided into 254 steps. This identifies the number of straight lines executed, the number of steps as multiples of 254 executed and the number of steps executed presently in the incremented step. These data are sent back to the P.C, which back calculates using the data received and the coordinates of the edges selected stored in the P.C.'s memory the location of the obstacle. The step by step logical approach to the above discussion is given in the sections below.

## 3.3 SOFTWARE - P.C BASED

### 3.3.1VISIBILITY METHOD

1. Accept the number of obstacles in the space

2. Accept the coordinates (x & y) of the obstacles present, start and the end (point Ø and n+1 respectively).

3. Identify the number of diagonals of the above mentioned obstacl

4. From the point Ø (start point) check for visibility (intersect with diagonals) with each of the other points. Enter

distance between the two points in question in the matrix $A_{ij}$ if continuty is there otherwise enter a high value to signify discontinuty. The adjacency matrix A will be symmetric.

5. Check if the next point lies in the object j or not (considering the no of edges) if not take the next point from the next object.

6. Repeat this till all the points are covered.

3.3.2<u>DIJKASTRA'S ALGORITHM</u>:

1. Take the continuity matrix A.

2. Give initially high value to all station except the start which is the present permanent station.

3. For each of the remaining station reindex the station to retain the minimum of the present and the sum of the present permanent station's index and the distance ($A_{ij}$) between the two.

4. Identify the minimum of all the indices (except the permanent station's ).

5. Reinitiate the new permanent station as the node found from the above.

6. Identify the permanent station as in (2) as the forerunner of the station identified in (5).

7. Repeat (2) through (6) till the present permanent station as identified in (5) is the goal point.

8. Retrace the point going from goal point to the start point from the matrix as entered in (6).

9. Connect these points using straight lines and find out the number of steps for the two motors required to reach the

points.

10. Send these data to the microprocessor using the COM1 port of the P.C.

### 3.3.3 LOCATION RETRIVAL:

1. Read the data inputed from the microprocessor.
2. Read the via points from the file where it is stored.
3. Identify the major station crossed.
4. Calculate the position of interrupt by computing the total number of steps executed in the straight line segment.
5. Accept new points, keeping the major stations after interrupt and location of interrupt intact.
6. Replan the path according to straight line trajector if the path earlier planned is model based or with cubic planning if explicit.
7. Calculate the amount the wheel axis has to moved back and the other data for executing the planned path.
8. Transmit the data as before.

## 3.4 SYSTEM SOFTWARE (8085 BASED):

### 3.4.1 CONTROLLER PROGRAM:

1. Initiate the 8251 and 8255.
2. Accept data from P.C by poling the port and checking for the receive buffer full status.
3. Come out of the poling mode once the transfer is over (identified by a set of zero's).
4. Start by loading a given memory location (FE00 &FE02) with the location of the start of the stack (FA00 & FB00).

5. Check if present value at FAØØ stack is zero. If zero then increment the stack pointer (FAØØ series) and service the motor controlling the angular change of the wheel shaft.

6. Once the identifier (set of zeros ) is found come out of execution.

3.4.2 <u>SERVICE ROUTINE FOR MOTORS</u>:

1. Load the registers with the number of steps, steps and the speed of the motor.

2. The first data from the FAØØ stack is the number of steps and the second for the direction.

3. Load the DE register pair with the count . down value(pertaining to the delay).

4. Call the software delay with a time of (24* N+17).

5. Decrement the counter identifying half.clock pulse and output data (Ø or 1) complementing each time.

6. Repeat (4) and (5) and decrement register containing the number of steps.

7. Repeat (3) through (6) till the value in the register for number of steps is zero.

8. Repeat (1) through (7) till all the steps are executed.

3.4.2 <u>LOCATION RETRIVAL</u>:

1. Identify the location of pointer in FA** and FB** stack from the memory location FEØØ and FEØ2 respectively.

2. Load this location on to HL register and then the L content to some register.

3. Deduct 2 from A and divide it by 2. This gives the number of steps (multiples of FE) executed including that of the present

countdown.

4. Starting from FAØØ count the number of zeros (first zeros only) till we reach the present location of FAØØ stack. Enter the location of the location of the zeros in FEØ4-Ø5, updating each time zero is found.

5. Output the value counted in (4).(This gives the number of straight lines completed.

6. With the last location of zero find out the number of steps executed and output this (this gives the number of FE steps executed in the present line)

7. Output finally the present step being executed.

# CHAPTER - 4

## RESULTS OF EXPERIMENTS

Any mechanical system having a number of moving parts invariably poses some operational and functional problems. This, in many cases, can be easily rectified by having a close loop system monitoring the desired output and the actual output. The stepper motors are controlled using an open loop controller wherein it is expected that a change in sequence of the coil energisation would cause the motor to rotate by 1.8 $^{o}$. Thus the wheel shafts will also rotate by 1.8$^{o}$. This is not the case and errors are introduced in the angular rotation of the wheel shafts. An obstacle filled environment was layed out for the vehicle to manoeuver in (fig(15) ). The P.C selected the edges to act as via points and the vehicle moved based on the data received from the P.C. The vehicle was made to trace and retrace the path. The shift in the actual position of the wheel (in both the r and $\theta$ coordinates) was measured to check for the accuracy and repeatability of the vehicle. The possible explanation for the errors could be the sudden acceleration/deceleration of the motor bringing in a lot of jerk in the vehicle. This also increases the backlash errors. The error is not very large and can be easily adjusted for in the program. This problem can be eliminated by using servo controlled D.C. motors. The response is tabulated below. The total distance traversed was about 1.5 meters each way.

| No. | Forward Direction deviation |  | Reverse Direction deviation |  |
|---|---|---|---|---|
|  | Radial | Angular | Radial | Angular |
| 1 | 5 mm. | $\approx 1^{o}$ | 5 mm. | $\approx 1^{o}$ |

| No. | Forward Direction deviation |  | Reverse Direction deviation |  |
|---|---|---|---|---|
|  | Radial | Angular | Radial | Angular |
| 2 | 5 mm. | $\approx 1^{\circ}$ | 1 mm. | $\approx 1^{\circ}$ |
| 3 | 5 mm. | $\approx 1^{\circ}$ | 2 mm. | $\approx 1^{\circ}$ |
| 4 | 5 mm. | $\approx 1^{\circ}$ | 5 mm. | $\approx 1^{\circ}$ |
| 5 | 7 mm. | $\approx 1^{\circ}$ | 5 mm. | $\approx 1^{\circ}$ |
| 6 | 5 mm. | $\approx 1^{\circ}$ | 5 mm. | $\approx 1^{\circ}$ |
| 7 | 10 mm. | $\approx 1^{\circ}$ | 5 mm. | $\approx 1^{\circ}$ |
| 8 | 10 mm. | $\approx 1^{\circ}$ | 5 mm. | $\approx 1^{\circ}$ |

**Speed of traverse maintained at 2 m/min.**

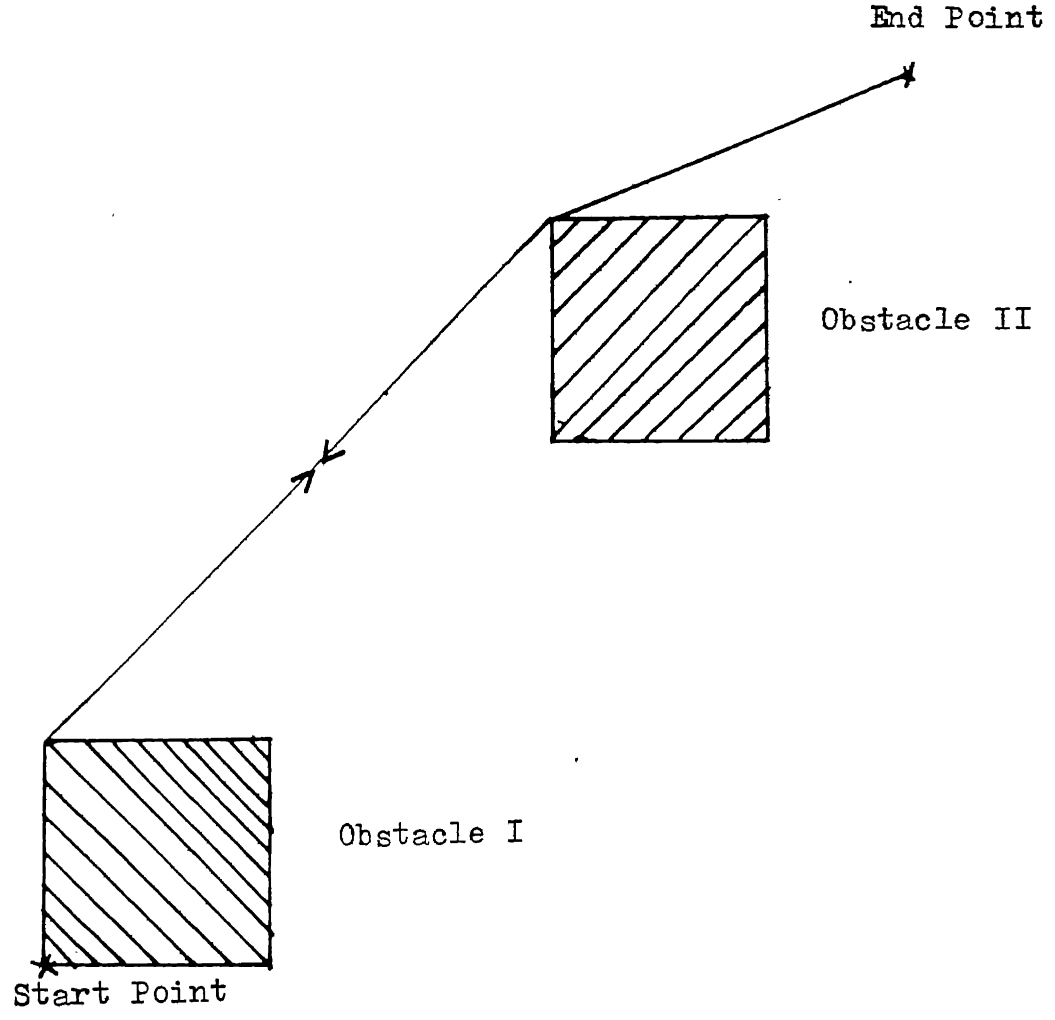


Fig. 15    Layout of Obstacles

# CHAPTER - 5

# SCOPE FOR FURTHER WORK

## 5.1 INTRODUCTION:

The vehicle system developed here does not fully satisfy the ever growing need for a complete self-contained mobile robot system. Amongst the various developmental work that could be done woiuld be an all contained system which does not depend on the user all the time for inputs. The vehicle ideally should have an unlimited range . These two problems can be easily overcome by having

1. A two way communication between the station and the vehicle and

2. A vision system which could help the robot to feel its environme along with some range finders.

When simulating the system for location retrival we assume the user to take control the moment some new obstacles come in the 'Obstacle Free Zone' identified earlier by the host program. This however is not always needed and the system could be programmed to select its own options instead. These two suggestions are dealt with in a little more detail below.

### 5.2.1 COMMUNICATION SYSTEM BETWEEN STATION AND ROBOT:

The link between the mobile robot and the base station has to be some reliable two way communication system like a radio system. This link can be established in one of the two ways.

1. Converting the data from the host P.C into radio signals or
2. Converting the data from the stepper motor controller and the

SIR-1's controller into radio signals. In both the cases we need a system similar to the 'MODEM' which modulates the digital signal from the controller and converts them into analog signal and receives it by reconverting it to digital. In such a system the 'Remote Controlled Vehicle' should have a power pack which should supply the required power for the manipulator's and the vehicle's motors. The power pack will hence be heavy and so the motors on the vehicle perhaps need to be changed. Since closed loop system have better control characteristics D.C. Servo motors should be preferred. With such a motor we would be able to bring about gradual acceleration \ deceleration of the vehicle

### 5.2.2 VISION SYSTEM:

The present model uses an user interrupted software which is tiresome when continuous replanning is necessary. This hurdle can be eliminated by using a vision system supported with range sensors to sense and recognize the surroundings. The vision would do a lot to bring about accurate path planning.

## APPENDIX

## MATHAMETICAL ANALYSIS

The software uses either explicit or model based programming for sending data to the microprocessor. In explicit programming, the path is fixed considering 3 points at a time starting with the first point. The curve, passing through these points, are fixed considering that the slope of intermediate points (i.e. all except the first and the last) are decided by the fraction of the slope of the lines joining its neighbouring points. The continuity is maintained and the equations

$$X = A1*T^3+B1*T^2+C1*T+D1;$$

$$Y = A2*T^3+B2*T^2+C2*T+D2;$$

are used to fix the intermediate points in the curve. The constants in the equations are evaluated using the conditions of continuty at the two end points and the slopes at the edges. The curves are fitted piecewise.

The visibility method (model based planning) works as explained below:

Consider a point I with coordinates X(I),Y(I) and another point J with coordinates X(J),Y(J). The line joining these two points can be represented as

$$X = X(I) + L1*[X(J)-X(I)];$$

$$Y = Y(I) + L1*[Y(J)-Y(I)];$$

while the equation of the line joining the vertices of a convex polygon forming its diagonal can be represented as

```
X = X(1) + L2*[X(2)-X(1)];
Y = Y(1) + L2*[Y(2)-Y(1)];
```

The coordinates X and Y will be the same in case of an intersection and so the equations can be solved to find the point of intersection. The non dimensional parameters L1 & L2 lie betweem Ø and 1. Thus if intersection takes place then the values of L1 and L2 on solving the above set of equation will be within Ø and 1.

## APPENDIX - 2

## COMPLEXITY OF COMPUTATION

1. VISIBILITY METHOD:

Let the number of diagonals be D ( fraction of the number o objects). Let the number of objects in the environment be N (fractio of the number of points M). The visibility approach searches fo continuity between any two points. Thus this can be brought about wit ${}_nP_2$ permutations i.e., n*(n-1) /2 possibilities. In other words maximum of n*(n-1) non identical lines can be drawn. Each of thes lines are checked for intersection with the diagonals. So for eac line on an average 'D' diagonals have to be checked. The maximum o the total number of checkings doneis n*(n-1)*n/2. Thus the complexi of computation is $(n^3)$. However since we come out of the loop wh there is intersection before end of search the number of calculatio is reduced.

2. DIJKASTRA'S ALGORITHM:

Consider 'N' points with the n * n continuty matrix A[i,j The first stage in the program assigns the indices and reallocates n indices to the various points. Making the start station as permanen we find the minimum of the indices computed for (n-1) points. This repeated for a maximum of (n-1) points and in the worst case complexi O(N-1)*(N-1)complexity results. Typically once the permanent stati is identified the node is no more involved in the computation. In t worst case complexity of $O(N^2)$ results. The sorting algorithm (bubb sort) gives us the complexity of O(N log N).

## ASSEMBLY LANGUAGE PROGRAM

The assembly level program for the 8085 processor is divided into three divisions (1) 8251A programme with the data receive section (2) 8255 part with the stepper motor controlling signal & (3) the interrupt portion.

A      THIS PART IS FOR 8251A PROGRAMMING AND FOR RECEIVING DATA

```
------------------------------------------------------------------------
ADDRESS             MNEONICS              OPERATION CODE
------------------------------------------------------------------------

FC00                 3E 00                   MVI A,00
FC02                 D3 FF
                     D3 FF
                     D3 FF
                     3E 40
                     D3 FF
                     3E CA
                     D3 FF

                     3E 37
                     D3 FF
FC14                 2A 00 FE                LXLD FE00
FC17                 CD FC50                 CALL FC50
FC1A                 C3 60 FC                JMP FC60
                       00
                       00
FC1F                   77                    MOV M,A
                       23                    INX H
```

```
FC21          0E 00          MVI C,00
              CD 50FC        CALL FC50
FC26          DB FE          IN PORT FE
              77             MOV M,A
              23             INX H
              22 00 FE       SHLD FE00
FC2D          2A 02FE        LXDL FE02
              CD 50FC        CALL FC50
              DB FE          IN PORT FE
              77             MOV M,A
             23             INX H
              22 02FE        SHLD FE02
FC3A          C3 14FC        JMP FC14
FC3D          76             HALT
FC3E          77             MOV A,M
              23             INX H
              3E 03          MVI A,03
              0C             INR C
              B9             CMP C
              CA 4BFC        JZ FC4B
              C3 23FC        JMP FC23
FC4A          76             HALT
FC4B          C3 69FC        JMP FC69
FC50          DB FF          IN PORT FF
              E6 01          ANI 02
              CA 50FC        JZ FC50
FC57          C9             RETURN
FC60          DB FE          IN PORT FE
```

```
                B7              OR A
                CA 3EFC         JN FC3E
FC65            C3 IFFC         JMP FCIF
```

PROGRAM FOR 8255 AND STEPPER MOTOR CONTROLLING PART

```
ADDRESS         MNEONICS              OPERATION CODE
FC69
FC70             3E 80          MVI A,80
FC72            D3 03           OUT 03 PORT
FC74            2A 00 FE        LXDL FE00
FC77            46              MOV B,M
FC78            23              INX H
FC79            4E              MOV B,M
FC7A            23
FC7B            22 00 FE        SHLD FE00
                2A 02 FE        LDHX FE02
FC81            56              MOV D,M
                23              INX H
                22 02 FE        SHLD FE02
                78              MOV A,B
                B7              OR A
                CA FCB0         JZ FCB0
                3E F0           MVI A,F0
                81              ADD C
                4F              MOV C,A
FC8F            2E 01           MVI L,01
FC91            CD 00 FD        CALL DELAY FD00
FC94            D3 01           OUT PORT 01
```

```
FC96          FB            ENABLE INT.

              2D            DCR L

              F2 91FC       JP FC91

FC9B          05            DCR B

              C2 8FFC       JNZ FC8F

              IE 00         MVI E,00

FCA1          C3 70FC       JMP FC70
```

<u>SERVICE ROUTINE STARTS FOR THE JUMP FCB0 HERE</u>

```
ADDRESS       MNEONICS      OPERATION CODE

FCB0          1C            INR E

              2A 00FE       LXLD FE00

              46            MOV B,M

              78            MOV B,A

              B7            ORA

              CA E0FC       JZ FCE0

              23            INX H

              4E            MOV C,M

              23            INX H

              22 00FE       SHLD FE00

              2A 02FE       LXLD FE02

              56            MOV D,M

              23            INX H

              22 02 FE      SHLD FE02

              3E F0         MVI A,F0

              81            ADD C

              4F            MOV C,A

FCCC          2E 01         MVI L,01
```

```
FCCE            CD 00 FD        CALL DELAY FD00
                D3 00           OUT PORT 00
FCD3            FB              ENABLE INT.
                2D              DCR L
                F2 CEFC         JP FCCE
FCD8            05              DCR B
FCD9            C2 CAFC         JNZ FCCA
FCDC            C3 70 FC        JMP FC70
------------------------------------------------------------------

ADDRESS         MNEONICS        OPERATION CODE

FCE0            1C
                23
                23
FCE3            22 00FE         SHLD FE00
                2A 02FE         LXDL FE02
                23              INX L
FCEB            22 02FE         SHLD FE02
                3E 04           MVI A,04
                BB              CMP E
                CA FFFC         JZ FCFF
FCF4            C3 70FC         JMP FC70
FCFF            76              HALT
FD00            F3              DIS INTERRUPT
FD01            C5              PUSH B
                D5              PUSH D
FD03            62              MOV H,D
FD04            11 FF00         LXI D 00FF
```

```
FDØ7          CD BCØ3          CALL DELAY Ø3BC

FDØA          25               DCR H

              C2 Ø4FD          JNZ FDØ4

FDØE          79               MOV A,C

FDØF          EE FO             XRI FO

              DI               POP D

              C1               POP B

FD13          4F               MOV C,A

FD14          C9               RETURN

FD15          76               HALT
```

---

THIS PART IS FOR THE INTERRUPT ROUTINE

```
ADDRESS        MNEONICS          OPERATION CODE

FD2Ø             2A ØØFE          LXDL FEØØ

FD22             7D               MOV A,L

                 16 ØØ            MVI D,ØØ

                 DE Ø2            SBI Ø2

                 ØF               RRC

                 DE Ø1            SBI Ø1

                 4F               MOV C,A

                 ØØ               NO OPERATION

                 21 ØØFA          LXI FAØØ

FD3Ø             7E               MOV A,M

                 B7               OR A

                  CA 5ØFD          JNZ FD5Ø
```

```
A                MOV A,D
    FD3C              D3 FF                 OUT PORT FF
                      7D                    MOV A,L
    FD3F              2A 04FE               LXDL FE04
                      4D                    MOV C,L
                      91                    SUB C
    FD44              DE 04                 SBI 04
                      0F                    RRC
                      D3 FF                 OUT PORT FF
                      78                    MOV A,B
    FD4A              D3 FF                 OUT PORT FF
    FD4C              C3 00FC               JUMP FC00
    FD50              22 04FE               SHLD FE04
                      23                    INX H
                      23                    INX H
                      14                    INR D
                      C3 35FD               JMP FD35
    FD59              76                    HLT
```

## REFERENCES

1 Muller Ing Thomas,"Automated Guided Vehicles",Ellis Horwood Limited, England, 1984.

2 Thring, M.W. "Robots and Telechirs", IFS (Publications) Ltd., Springer-Verlag, New York, 1984.

3 Bhagirathsinh N Gohil (1987). "Development Of Teleoperated Vehicle",M.Tech Thesis, I.I.T.,Kanpur 1987

4 Lozano-Perez,T., and Wesley,M.A.(1979). "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles,"Comm. ACM, vol. 22,no. 10 pp. 560-570.

5 Maitra,G.L. "Handbook of Gear Design",Tata Mc.Graw-Hill New Delhi, 1986.

6 Shigley, J.E."Mechanical Engineering Design" 4/e,McGraw Hill New York.

7 Donaldson "Tool Design", Tata Mc.Graw Hill, New Delhi, 1986.

8 ASTME Handbook of Tool Design", Prentice Hall of India, 1984.

9 G. Yates Fletcher, David P Mc. Allister (1986). "Natural Bias Approach to Shape Prescribed Curves," CAD 86

Jan/Feb, pp.48-52.

10 Papalintriou and Steiglitz.K (1982). Combinatorial Optimization - Algorithms & Complexity, Prentice Hall Inc., New Jersey.

11 Dijkstra. E. (1959). "A Note on Two Problems in Connection with Graphs, "Numerische Mathematik, vol. 1, pp.269-271.

12 Rich,E (1983). Artificial Intelligence, McGraw-Hill, New York

13 Udupa Shriram, M. (1977). "Collision Detection and Avoidance in Computer Controlled Manipulators ," Proceeding of the International Conference on Artificial Intelligence - 5, 1977, pp. 737-748.

14 Vinytics Microprocessor User's manual, Vinytics, New Delhi.